



FORMAT DELS FITXERS VECTORIALS ESTRUCTURATS (TOPOLÒGICS O NO) DEL MIRAMON (PUNTS, ARCS, NODES I POLÍGONS)

Autors del document: Abel Pau i Xavier Pons
Proposta inicial: 18-09-2014 (04-12-2014 v. 1.1)
Darrera modificació i versió: 25-04-2024. 2.2

1. ANTECEDENTS, MOTIVACIÓ I ESTRUCTURA D'AQUEST DOCUMENT.....	2
2. DESCRIPCIÓ DEL FORMAT DELS FITXERS VECTORIALS ESTRUCTURATS DEL MIRAMON	3
2.1. Capçaleres estructurades comunes a tots els fitxers	5
2.1.1. Capçalera comuna a tots els fitxers vectorials estructurats	5
2.1.2. Altituds per al cas de fitxer 3D.....	9
2.2. Fitxer de punts .pnt	12
2.3. Fitxer d'arcs .arc.....	13
2.4. Fitxer de nodes .nod.....	16
2.5. Fitxer de polígons .pol	18
2.6. Resum del format per a tots els tipus de fitxers.....	23
3. ALGUNES CONSIDERACIONS SOBRE ELS FITXERS D'ARCS I POLÍGONS	24
4. EXEMPLE IL·LUSTRATIU DE POLÍGONS COMPLEXOS	25

1. Antecedents, motivació i estructura d'aquest document

Als inicis del MiraMon, Xavier Pons i Joan Masó van dissenyar i crear uns formats per als fitxers vectorials del programa, els quals havien de poder contenir Punts, Arcs, Nodes i Polígons, i fer-ho d'una manera equilibradament entrellaçada per tal de maximitzar-ne la coherència (reduir potencials inconsistències a través d'utilitzar elements comuns entre ells) però sense comprometre'n el rendiment (quelcom que hauria passat en un model totalment normalitzat d'interdependències). La seva concepció perseguia, doncs, que els nous fitxers tinguessin una estructura que permetés un tractament informàticament molt àgil i, alhora, que permetessin explicitar les relacions topològiques entre les entitats o objectes geogràfics quan fos necessari.

Aquests fitxers tenen una **estructura informàtica** interna que els fa aptes per a l'**accés eficient d'informació vectorial 2D i 3D** (des de la v. 1.1). És per això que s'utilitzen també per a contenir informació vectorial sense estructura topològica. Per tant, és important entendre que:

1. **El format vectorial estructurat del MiraMon pot contenir fitxers amb o sense estructura topològica explícita.**
2. **El format recull, com es veurà, informació sobre si conté elements vectorials dels quals es coneix les relacions topològiques, o no (cas conegut popularment com a "espaguetis" en el cas d'elements lineals).**
3. **Un fitxer estructurat del MiraMon pot contenir dades importades des d'un fitxer sense topologia (com per exemple un SHP o un DXF) i en aquest cas serà, simplement, un fitxer estructurat, NO un fitxer topològic.**
4. **Un fitxer estructurat del MiraMon pot contenir dades amb estructura topològica explícita (per exemple resultant de digitalització topològica en el MiraMon, resultant d'un procés d'estructuració topològica amb certes opcions d'aplicacions com el LinArc, o resultants d'una importació d'un format sense topologia, com un SHP, en què durant la importació s'ha aprofitat per construir la topologia). En aquest cas serà, pròpiament, un fitxer estructurat topològic.**

En la v. 2.0 d'aquest document es recull la v. 2.0 del format, que essencialment admet valors de 64 bits, i ja no de 32, per als enters que indiquen posicions dintre del fitxer (*offsets*) i per als identificadors dels objectes, així com alguna altra modificació. Això fa que **a partir de la v. 2.0 tant la grandària possible dels fitxers, com el nombre d'elements que poden contenir, són virtualment infinits**.

Aquest document complementa la informació del fitxer d'ajuda del MiraMon "*Descripció dels formats vectorials del MiraMon*", on es dóna la descripció dels formats estructurats des del punt de vista de l'usuari. Es recomana la seva lectura si no s'està familiaritzat amb els fitxers del MiraMon. S'hi pot accedir

directament a través de l'enllaç de l'ajuda online del MiraMon:
<https://www.mirammon.cat/help/cat/mm32/ap2.htm>

En l'[apartat 2](#) d'aquest document es descriu la part comuna (capçalera topològica) a tots els fitxers i després la part específica per a cada tipus de fitxer. En l'[apartat 3](#) es fan algunes consideracions sobre arcs i polígons, i en l'[apartat 4](#) s'il·lustra un cas complex de polígon.

És important entendre que, a diferència d'altres models d'estructura topològica, en què els arcs només es poden utilitzar per a ciclar una única capa de polígons, en el model del MiraMon una mateixa capa d'arcs pot ser utilitzada per a ciclar diverses capes de polígons (una capa de límits administratius pot servir per a ciclar una capa de municipis, una capa de comarques, etc); naturalment, en cada capa de polígons només s'utilitzen aquells arcs necessaris per a ciclar els corresponents polígons. Això fa que la clàssica informació topològica sobre quin polígon és a la dreta o a l'esquerra de cada arc no està en la capa d'arcs, sinó en la pròpia capa de polígons.

2. Descripció del format dels fitxers vectorials estructurats del MiraMon

Una capa vectorial del MiraMon, ja sigui de Punts, Arcs, Nodes o Polígons consta de diversos fitxers. Els fitxers principals sempre són 3:

- **Un fitxer que conté la part gràfica** (geomètrica i, si és el cas, topològica), amb coordenades, dependències espacials, etc, i que tenen extensió **.pnt**, **.arc**, **.nod** i **.pol**. En aquest document s'explica el format dels fitxers corresponents a la part dels elements gràfics.
- **Una taula principal d'atributs d'usuari**, que està en format DBF, o en format DBF estesa si es requereixen més de 254 camps, camps de més de 254 caràcters, noms de camps de fins a 128 caràcters, un nombre de registres superior a 4200 milions, etc. El format DBF és un format ben conegut i documentat, i el format DBF estesa es pot trobar també documentat al document tècnic [Especificació del format "DBF estesa"](#) creat per Xavier Pons i Abel Pau. **L'extensió .dbf va precedida de les lletres 'T', 'A', 'N' o 'P', segons es tracti d'una DBF relativa a una capa de punts, arcs, nodes o polígons.** La resta del nom és igual a la del fitxer gràfic i s'emmagatzema al mateix directori.
- **Un fitxer de text, en format INI del Windows**, descrit a l'ajuda del MiraMon, que **conté les [metadades de la capa](#)** i també descriu tant **la simbolització per defecte** (opcional) com les eventuais **relacions de la taula principal amb altres taules** (siguin DBF o altres, com taules o consultes en fitxers Access, grans gestors de bases de dades com SQL Server, Oracle, etc). **L'extensió per a aquest fitxer és .rel, i va precedida de les lletres 'T', 'A', 'N' o 'P', segons es tracti d'un REL relatiu a una capa de punts, arcs, nodes o polígons.** La resta del nom és igual al del fitxer gràfic i s'emmagatzema al mateix directori.

Internament els n punts, arcs, nodes i polígons s'indexen des de 0 (el primer element és el 0, el segon l'1, etc, fins als n-1 elements que conté el fitxer). Aquesta numeració proporciona el que en direm identificador gràfic. No està mai escrita en el fitxer binari i ve donada per l'ordre en què estan escrits els elements en el fitxer. En canvi, des del punt de vista de l'usuari, el MiraMon normalment mostra una numeració des d'1, que sembla més natural, excepte quan cal donar una informació altament tècnica. Per exemple, quan en una consulta per localització el MiraMon mostra "Element gràfic 3 de 8", internament es correspon a l'identificador gràfic 2. En els fitxers de polígons, el polígon universal o polígon zero és aquell que està "per fora de tots els altres polígons" (o en forats interiors quan aquest no tenen contingut de cap tipus), i, per tant, el primer element "funcional" és el polígon amb identificador 1; trobareu més informació sobre el polígon zero a l'[apartat 3](#).

Cal tenir en compte que l'ordre en què estan escrits els bits dels bytes sempre segueix la convenció Intel, no la Motorola. En aquest document el nom "double" fa referència a un nombre real de 64 bits (anomenat *double* en el llenguatge de programació C), que té prou precisió i rang numèric per a l'emmagatzematge de les coordenades emprades en informació geogràfica.

Comentari sobre offsets: A més, cal tenir en compte un comentari general en referència a qualsevol offset que aparegui en aquestes especificacions:

Fins la versió 1.1, inclosa, el valor de l'offset que documenta on anar a buscar un element (una coordenada X,Y, una coordenada Z...) estava guardat en una variable entera sense signe i de 4 bytes (el valor màxim que podia emmagatzemar era de 4 294 967 295), és a dir, que la primera coordenada Z del l'últim punt emmagatzemat podia estar, com a màxim, a la posició byte corresponent a una mica abans del final de la mida de fitxer d'aproximadament 4.29 Gbyte (en aquest document utilitzem els prefixos de mega (M), giga (G) o tera (T) com en l'SI, no com en informàtica: dividint entre múltiples de 1000, no de 1024).

A partir de la versió 2.0 aquest nombre es veu ampliat a 8 bytes, podent arribar a **fitxers de 18 milions de Tbyte**. Díficilment caldrà ampliar-lo més en vida nostra (o això devien pensar els qui creien que 4 Gbyte de memòria era "virtualment infinit", com els creadors del llenguatge C, que van preveure que la funció fseek() només necessitava un enter de 32 bits per a indicar un offset). El mateix passa amb els **identificadors dels objectes**: passen a ser codificats amb un enter de 64 bits sense signe i, per tant, **el nombre d'objectes possible és virtualment infinit**.

2.1. Capçaleres estructurades comunes a tots els fitxers

2.1.1. Capçalera comuna a tots els fitxers vectorials estructurats

Tots els fitxers vectorials estructurats del MiraMon tenen una part comuna a l'inici: la capçalera topològica. Aquesta capçalera té una de grandària **sizeof_TH**, que val **56 bytes (48 bytes en la versió 1.x)**. L'estructura i el contingut de la resta del fitxer depèn de si és de punts, arcs, nodes o polígons.

Descripció dels sizeof_TH bytes de la capçalera:

Topo Header (TH)		
0	3	File type (PNT, ARC, NOD, POL)
3	2	Version ("0"- "99")
5	1	". "
6	1	Subversion ("0"- "9")
7	1	Flag (1 byte)
8	8	Bounding box: Minimum X
16	8	Bounding box: Maximum X
24	8	Bounding box: Minimum Y
32	8	Bounding box: Maximum Y
40	8 (4 en la v. 1.x)	Element count
48	8 (4 en la v. 1.x)	Reserved

TH+44 en la v. 1.x

File type, Version, Subversion (Tipus de fitxer, Versió, Subversió)

Es tracta d'una cadena formada per **3 caràcters** que declaren el tipus de fitxer (PNT, NOD, ARC o POL i que coincideix amb l'extensió del fitxer) i per **4 caràcters** que diuen la versió del format, amb una xifra decimal; aquests 4

caràcters s'alineen a la dreta. Per exemple, un començament tipus és: "PNT 1.1". Si un dia existís en la versió 12.3 hi diria: "PNT12.3".

En la versió 1.x el nombre de bytes destinats a indicar el nombre (enter) d'objectes és 4 i, per tant, es poden desar fins a 2^{32} elements (~4200 milions d'elements).

A dia 15 d'abril de 1997 es va dissenyar una versió 1.1 per a suportar coordenades 3D (X,Y,Z) que guardava fidelment la compatibilitat descendent amb la inicial 1.0, que només suportava coordenades 2D (X,Y).

A dia 21 de març del 2023 es dissenya una segona versió, 2.0 que guarda fidelment la filosofia de la 1.1, però ja no n'admet la lectura per part de les aplicacions compilades sense conèixer les especificacions de la 2.0. Aquesta versió sorgeix de la necessitat d'ampliar tant el nombre d'elements gràfics que pot contenir un fitxer, com la ubicació de les coordenades d'aquests objectes dins dels fitxers; aquests valors que es codifiquen en enters de 8 bytes (64 bits) sense signe. Com que s'usen 8 bytes, en la versió 2.0 el nombre total d'elements possibles és de 2^{64} elements (~18 milions de Tbyte).

Més endavant en aquest document s'explica com accedir a la ubicació de les coordenades per a les versions 1.x i 2.0.

Flag (bandera)

El **byte flag** pot definir fins a 8 propietats lògiques en sengles bits (Veritat:1 o Fals:0) amb significats diferents segons la natura geomètrica dels objectes del fitxer. En la data d'aquest document estan definits els següents bits.

Bits vàlids per a fitxers PNT, ARC, NOD i POL:

bit 0

Indica que la topologia ha estat verificada per alguna aplicació que mereix prou confiança. Així, 1 informa que el fitxer ha estat generat amb una aplicació pròpia del MiraMon que garanteix que la topologia indicada és correcta, o que el fitxer ha estat importat des d'un altre format on les relacions topològiques eren presents i es consideren fiables.

bit 1

Indica que el fitxer ha estat generat amb una aplicació del MiraMon. Noteu que aquest bit pot estar a 1 encara que el fitxer NO continguin topologia en el moment actual.

Bit només vàlid per a fitxers PNT, ARC i POL (en fitxers NOD ha de valer 0 en les versions 1.x i 2.0).

bit 2

Per a PNT: Un valor 1 indica que el fitxer de punts ve d'un fitxer POL via l'aplicació de suport al MiraMon (MSA) "[Etiqueta](#)" i conté una etiqueta sobre el polígon zero. La MSA del MiraMon "[AtriTop](#)" consulta això per heretar o no atributs del polígon zero. En cas de dubte escriu un 0 en el bit.

Per a ARC: Un valor 1 indica que el fitxer d'arcs conté només vores de polígons. Això vol dir que ha estat possible un procés de ciclat total (en el qual intervenen tots els arcs), el que garanteix que el fitxer no presenta ni nodes finals ni arcs amb el mateix polígon als dos costats (*halvers*). En cas de dubte escriuiu un 0 en el bit.

Per a POL: Un valor 1 indica que el fitxer ha estat etiquetat correctament amb la MSA del MiraMon "[AtriPol](#)" o "[AtriTop](#)", o amb una altra MSA del MiraMon com ara "[RasTop](#)". Concretament vol dir:

- No hi ha etiqueta en el polígon zero.
- No presenta reetiquetat incoherent dels polígons.
- No hi ha polígons sense etiquetes.

En cas de dubte o en el cas NOD escriuiu un 0 en el bit.

bit 3

Bit només vàlid per a fitxers PNT i POL:

Per a PNT: Un valor 1 indica que el fitxer de punts prové d'un fitxer POL via la MSA del MiraMon "[Etiqueta](#)" i **no** té una etiqueta sobre el polígon zero.

Per a POL: Un valor 1 indica que el fitxer de polígons conté grups (o regions) en polígons diferents del polígon zero. Si la topologia no està verificada cal que el bit 0 estigui apagat.

En cas de dubte escriuiu un 0 en el bit.

bit 4

Només per a PNT i ARC: El fitxer presenta coordenades 3D. Els fitxers POL i NOD poden ser 3D, però les seves coordenades Z sempre estan contingudes en el corresponent fitxer ARC.

bit 5

Aplicable a polígons, un valor 1 significa que els arcs que intervenen en el ciclat (recordeu que s'admet que hi hagi arcs no usats) només s'usen un cop. No es pot combinar amb el bit 0 ja que aquesta darrera bandera implicaria que els arcs s'usarien dos cops (si més no contra el polígon zero) i que les superposicions estarien prohibides, situacions que volem admetre en el cas dels polígons explícits (no topològics). Vegeu també la "[Nota sobre polígons explícits](#)" al final del document.

bit 6

Per a POL: Un valor 1 indica que el fitxer de polígons conté grups (o regions) al polígon zero. Això es pot interpretar com que algun altre polígon diferent del polígon zero té un o més forats al seu interior. Aquest bit juntament amb el bit 3 permet saber tota la informació referent a grups en un fitxer de polígons. Si la topologia no està verificada cal que el bit 0 estigui apagat.

En cas de dubte escriuiu un 0 en el bit.

Bounding box (envolupant)

Indica l'envolupant total en l'ordre minX, maxX, minY, maxY, igual que als fitxers REL de documentació (totes les envolupants dels fitxers binaris respecten aquest conveni). S'usa un valor *double* (real de 64 bits) per a cada membre de l'envolupant.

Element count (comptador)

Indica el nombre total d'entitats presents al fitxer. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Reserved (reservats)

Resten reservats per futures ampliacions o per a ús intern. Usos interns documentats:

- Durant operacions de modificació intensiva de l'estructura dels fitxers, com durant la digitalització o durant la importació d'altres formats:
 - El bit 0 del primer byte reservat valdrà 1 si les coordenades XY dels fitxers ARC (o, més rarament, PNT) no són a les zones després de AH (TH en punts) sinó en un fitxer ubicat al mateix directori, amb el mateix nom, però amb "extensió" "**#.~xy*" (on # és A per a capes d'arcs i T per a capes de punts). En aquesta situació, i en el cas de fitxers ARC/PNT, els offsets enregistrats a AH/TH es referiran a posicions en aquest fitxer "**#.~xy*", que seran reescrites a offsets en el fitxer ARC/PNT quan la modificació del fitxer es doni per finalitzada i la part de coordenades s'escriui en el fitxer ARC i el fitxer "**#.~xy*" sigui eliminat.
 - Per al cas de polígons, el bit 0 del primer byte reservat valdrà 1 si els índexs d'arcs dels fitxers POL no són a les zones després de PH sinó en un fitxer ubicat al mateix directori, amb el mateix nom, però amb "extensió" "**P.~idx*". En aquesta situació els offsets enregistrats a PH es referiran a posicions en aquest fitxer "**P.~idx*", que seran reescrites a offsets en el fitxer POL quan la modificació del fitxer es doni per finalitzada i la part d'índexs s'escriui en el fitxer POL i el fitxer "**P.~idx*" sigui eliminat.
 - Per al cas de nodes, el bit 0 del primer byte reservat valdrà 1 si els índexs d'arcs confluents a cada node dels fitxers NOD no són a les zones després de NH sinó en un fitxer ubicat al mateix directori, amb el mateix nom, però amb "extensió" "**N.~idx*". En aquesta situació els offsets enregistrats a NH es referiran a posicions en aquest fitxer "**N.~idx*", que seran reescrites a offsets en el fitxer NOD quan la modificació del fitxer es doni per finalitzada i la part d'índexs s'escriui en el fitxer NOD i el fitxer "**N.~idx*" sigui eliminat.

- El bit 1 del primer byte reservat valdrà 1 si les coordenades Z dels fitxers PNT o ARC no són a les zones després de TL o AL, respectivament, sinó en un fitxer ubicat al mateix directori, amb el mateix nom, però amb "extensió" "***#.~z**" (on # és T per a capes de punts i A per a capes d'arcs). En aquesta situació els offsets enregistrats a ZD es referiran a posicions en aquest fitxer "***#.~z**", que seran reescrites a offsets en el fitxer PNT o ARC quan la modificació del fitxer es doni per finalitzada i la part de coordenades s'escriu en els fitxers PNT o ARC i el fitxer "***#.~z**" sigui eliminat.

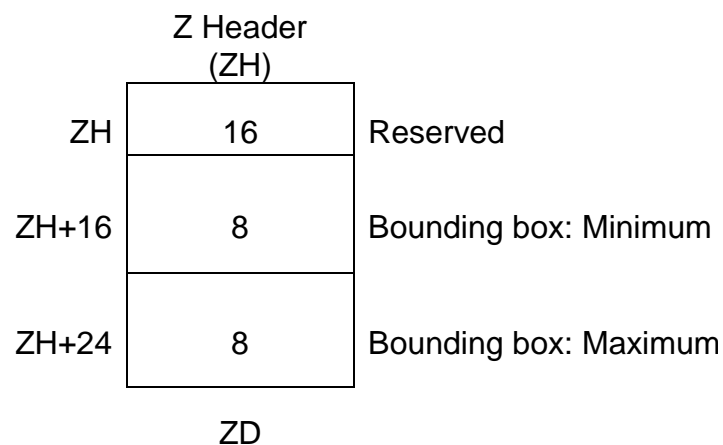
2.1.2. Altituds per al cas de fitxer 3D.

En el cas de PNT i ARC, si el fitxer és 3D, tindrem dues seccions per a definir les altituds.

Secció Z.

La secció Z es divideix en tres subseccions:

Subsecció ZH. Capçalera de **32 bytes** que es defineix a continuació.



Reserved

16 bytes que resten reservats per futures ampliacions. Ompliu amb 0.

Bounding box: Minimum Z

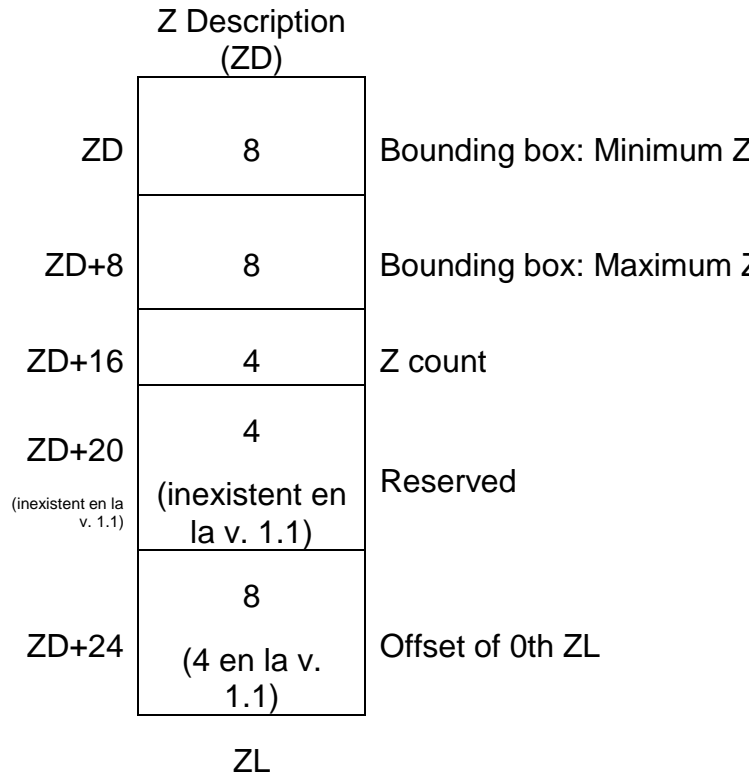
Mínim valor de totes les Z del fitxer. S'usa un *double*.

Bounding box: Maximum Z

Màxim valor de totes les Z del fitxer. S'usa un *double*.

Subseccions ZD.

Per CADA ELEMENT (punt o arc) s'escriu una capçalera de **32 bytes (24 bytes en la versió 1.1)** que té la següent estructura.



Bounding box: Minimum Z

Mínim valor de totes les Z del punt.

Bounding box: Maximum Z

Màxim valor de totes les Z del punt.

Z count

En el cas de PNT: . És el nombre d'altituds del punt i sempre és un valor negatiu o un valor zero. En la secció d'arcs es descriu què significa que "Z count" prengui un valor positiu (que en punts no té sentit).

En el cas d'ARC: Nombre d'altituds de l'arc. Si el **nombre d'altituds és positiu** indica el nombre d'altituds per a cada vèrtex de l'arc, i primer s'escriuen totes les altituds del vèrtex 0, després les de l'1, etc. Si el **nombre d'altituds és negatiu** indica el nombre d'altituds de l'arc, entenent que tots els vèrtexs tenen la mateixa altitud o altituds (cas d'una corba de nivell, per exemple). Useu -1.0E+300 com a sensedades (*NoData*) si no es coneix alguna de les altituds d'algun vèrtex (nota d'implementació en el programari MiraMon: aquest valor correspondrà al de la variable "const double NODATA_COORD_Z" de PrjMMVGl.h/c).

Nota sobre el sensedades a les Z: Arran de la recerca sobre fiabilitat en la comparació de reals de precisió doble feta al 2017 per Xavier Pons (i que forma part del document PrecisioEnReals_*.doc) es va proposar, **només per a la versió 2.0**, una "const double NODATA_COORD_Z_v_2" que es definís, com el -VALOR_ESTAD_INDEFINIT, a -2.9E+301 pels motius de comparació directa i impressió fiable explicats a la NOTA_XP_30-08-2017_30 del fitxer EstadStr.h del codi del MiraMon, quan es va precisament abandonar -1.0E+300 per al VALOR_ESTAD_INDEFINIT. Si en la lectura i escriptura la comparació fos fàcil i no calgués bifurcar fortament al codi, es podria adoptar el nou valor, però aquesta bifurcació allargaria la implementació d'aquesta v. 2.0 dels fitxers vectorials estructurats del MiraMon i, a més, a la pràctica l'antic valor -1.0E+300 no ha resultat de moment problemàtic per a les comparacions directes amb NODATA_COORD_Z (o sigui que no ha estat necessari utilitzar, en el codi del MiraMon les lentíssimes funcions de la família "double_diferents..."). Una altra possibilitat, molt millor, que es deixa aquí recollida però que tampoc no s'aplicarà encara en la v. 2.0, és la reserva d'un espai de 8 bytes per al sensedades en Z al format en disc, en els primers byte reservats del Z Header (ZH), i la càrrega d'aquest valor a l'estructura en memòria, de manera que, en llegir un fitxer en memòria hi hauria -1.0E+300 en les versions en què aquesta reserva no existeix (i s'assumeix el valor -1.0E+300) i en les versions que ja tinguin el sensedades escrit a aquesta posició del disc, en memòria hi hauria el que digui al disc, que per defecte hauríem escrit com a NODATA_COORD_Z_v_2. En el codi seria elegant perquè fariem com als ràsters, en què una membre de l'estructura especifica quin valor és el sensedades. En escriure s'hauria d'escriure el que digui el membre de l'estructura, que de nou si correspon a un fitxer 1.x o 2.0 seria NODATA_COORD_Z i, més endavant, NODATA_COORD_Z_v_2. Aquest pas l'acabarem fent en una versió posterior (potser una 2.x) per tal de poder explicitar valors de sensedades encara més fiables (que admeten una comparació amb un igual i que no tenen problemes en ser escrits els seus decimals) i fer desaparèixer problemes potencials tant en el codi del MiraMon com en al codi de les llibreries GDAL.

Com que en el model estructurat del MiraMon les vores dels polígons estan descrites per arcs, les explicacions d'aquest paràgraf apliquen a les vores de polígons. Similarment, els valors Z dels nodes es troben al fitxer ARC.

Exemple:

Un arc de 4 vèrtexs amb nombre d'altituds 2: s'escriuran les altituds del primer vèrtex, després les del segon, després les del tercer i finalment les del quart. Total: $4 \times 2 = 8$ altituds.

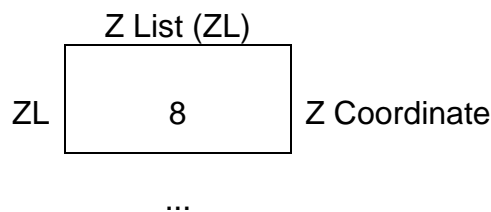
Un arc de 4 vèrtexs amb nombre d'altituds -2: s'escriuen les dues altituds de l'arc. Cada vèrtex tindrà aquestes dues altituds en el mateix ordre en tots els vèrtexs, tot i que només estaran escrites un cop.

Offset of 0th ZL

Indica l'offset on es troba escrita la primera coordenada Z (típicament una altitud). Només té sentit si el nombre d'altituds d'aquest punt és diferent de zero.

Subseccions ZL.

La secció ZL conté una llista d'altituds de cada punt o vèrtex d'arc.



Z Coordinate

Cas PNT: Altitud del punt que es representa. S'usa un double.

Cas ARC: Altitud del vèrtex indicat de l'arc que es representa o de tot l'arc. S'usa un double.

2.2. Fitxer de punts .pnt

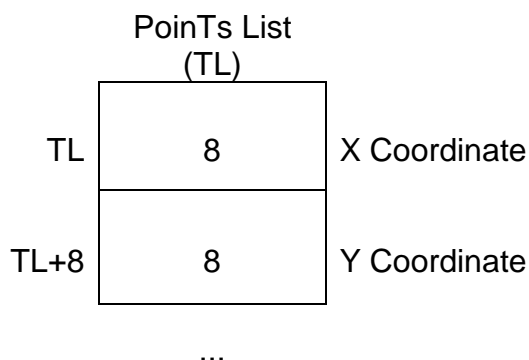
El format de fitxer de punts (PNT) conté dues seccions (tres en el cas 3D), descrites a continuació:

Secció TH. **Capçalera** comuna a tots els fitxers (**sizeof_TH**), anteriorment descrita. El camp "*File type*" correspon a la cadena "PNT".

Secció TL. Per a cada **PUNT** s'escriuen les coordenades del punt (**16 bytes**).

El primer punt està escrit en l'offset **sizeof_TH**, que és on acaba sempre la capçalera.

Descripció dels 16 bytes:



X Coordinate

Coordenada X del punt que es representa. S'usa un double.

Y Coordinate

Coordenada Y del punt que es representa. S'usa un *double*.

Quan el *flag* 4 de la secció "Capçalera topològica comuna a tots els fitxers" estigui activat (1), trobareu una secció 3, descrita a continuació.

Secció Z (s'aplica només en cas que el fitxer sigui 3D).

La descripció d'aquesta secció correspon a la secció Z de l'apartat "[2.1.2. Altituds per al cas de fitxer 3D](#)".

2.3. Fitxer d'arcs .arc

El format de fitxers d'arcs (ARC) conté tres seccions (quatre en el cas 3D), descrites a continuació:

Secció TH. **Capçalera** comuna a tots els fitxers (**sizeof_TH**), anteriorment descrita. El camp "*File type*" correspon a la cadena "ARC".

Seccions AH. Per a cada **ARC** s'escriu una capçalera de grandària **sizeof_AH**, que val **72 bytes (56 bytes en la versió 1.x)**. La primera capçalera està escrita en l'offset **sizeof_TH**, que és on acaba sempre la capçalera comuna TH. La resta de AH es troben a l'offset **sizeof_TH+sizeof_AH*id_arc**.

Descripció dels sizeof_AH bytes de la capçalera d'un arc:

Arc Header (AH)		
AH	8	Bounding box: Minimum X
AH+8	8	Bounding box: Maximum X
AH+16	8	Bounding box: Minimum Y
AH+24	8	Bounding box: Maximum Y
AH+32	8 (4 en la v. 1.x)	Element count
AH+40 <small>AH+36 en la v. 1.x</small>	8 (4 en la v. 1.x)	Offset of i-th AL
AH+48 <small>AH+40 en la v. 1.x</small>	8 (4 en la v. 1.x)	First node id
AH+56 <small>AH+44 en la v. 1.x</small>	8 (4 en la v. 1.x)	Last node id
AH+64 <small>AH+48 en la v. 1.x</small>	8	Length
(AL)		

Bounding box

Indica l'envolupant de l'arc descrit en aquesta capçalera en l'ordre minX, maxX, minY, maxY. S'usa un *double* per a cada membre de l'envolupant.

Element count

Indica el nombre total de vèrtexs de l'arc descrit en aquesta capçalera. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Offset of i-th AL

Offset al primer vèrtex de l'arc descrit en aquesta capçalera. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

First node id

Identificador del node inicial de l'arc. Aquest identificador fa referència a la llista d'identificadors de nodes que hi ha al fitxer de nodes associat al fitxer d'arcs. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Last node id

Identificador del node final de l'arc que fa referència a la llista de nodes del fitxer de nodes associat al fitxer d'arcs que es descriu. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Length

Longitud de l'arc que es descriu, en el mateix sistema de referència que les coordenades. S'usa un *double*. Convé indicar que les longituds sobre l'el·lipsoide també es calculen, però s'emmagatzemen a la taula principal (*A.dbf).

AL

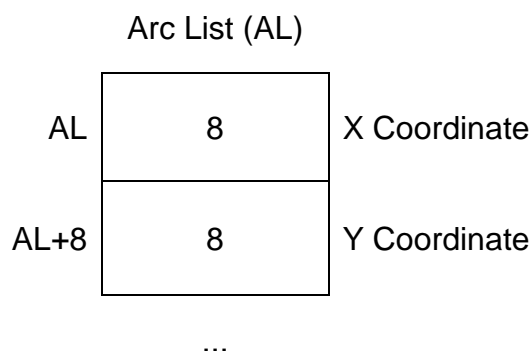
Immediatament després de la successió de AH correspondria que comencessin a aparèixer seccions AL, però no és necessari donat que cada secció AL es pot trobar a partir de l'offset indicat a la secció AH corresponent.

Secció AL.

Llista de coordenades dels arcs. Són les coordenades corresponents a cada arc individual a partir del nombre de vèrtexs de l'arc definit en el byte 32 del AH del corresponent *i* de l'offset definit en el byte 40 (36 en la v. 1.x) de la AH corresponent.

Per a cada **VÈRTEX** de l'arc s'escriuen les seves coordenades (**16 bytes**).

Descripció dels 16 bytes:



X Coordinate

Coordenada X del vèrtex que es representa. S'usa un *double*.

Y Coordinate

Coordenada Y del vèrtex que es representa. S'usa un *double*.

Quan el *flag* 4 de la secció "Capçalera topològica comuna a tots els fitxers" estigui encès (1) es trobarà una secció Z, descrita a continuació.

Secció Z (s'aplica només en cas que el fitxer sigui 3D)

La descripció d'aquesta secció correspon a la secció 1 de l'apartat "[2.1.2. Altituds per al cas de fitxer 3D](#)".

2.4. Fitxer de nodes .nod

El format de fitxers de nodes (NOD) conté tres seccions, descrites a continuació:

Secció TH. **Capçalera** comuna a tots els fitxers (**sizeof_TH**), anteriorment descrita. El camp "*File type*" correspon a la cadena "NOD".

Seccions NH. Per a cada **NODE** s'escriu una capçalera de grandària sizeof_NH, que val **12 bytes (8 bytes en la versió 1.x)**.

La primera capçalera està escrita en l'offset **sizeof_TH**, que és on acaba sempre la capçalera comuna TH. La resta de NH es troben a l'offset sizeof_TH+sizeof_NH*id_nod.

Descripció dels sizeof_NH bytes de la capçalera d'un node:

Node Header (NH)		
NH	2	Arcs count
NH+2	1	Node type
NH+3	1	Reserved
NH+4	8 (4 en la v. 1.x)	Offset of i-th NL
(NL)		

Arcs count

Indica el nombre total d'arcs que conflueixen en el node descrit. S'usa un *unsigned short int* (16 bits).

Node type

Indica el tipus de node. S'utilitza a partir de la versió 1.1. Els tipus de nodes possibles són: node típic (*Node type=0*), de línia (*Node type=1*), anell (*Node type=2*) i final (*Node type=3*). S'usa un *unsigned char* (8 bits).

Reserved

Resta reservats per futures ampliacions. S'usa un byte i pren el valor 0 actualment.

Offset of i-th NL

Offset al primer dels arcs que conflueixen al node descrit. S'usa un *unsigned __int64* (*unsigned __int32* en la versió 1.x). Els offsets cal que estiguin alineats a múltiples de 8 bytes. Així, un node d'anell (*Arcs count=1+reserved*) ocupa el mateix que un node de línia (*Arcs count=2* i no cal farcit). Per això la transformació d'un node d'anell a un de línia o viceversa no altera els offsets del fitxer.

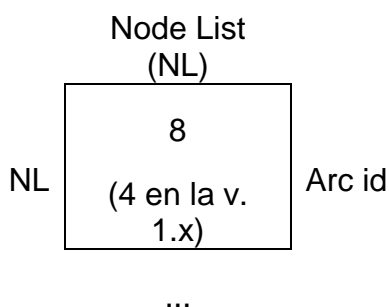
NL

Generalment, immediatament després de la successió de NH correspondria que comencessin a aparèixer seccions NL, però no és necessari donat que cada secció NL es pot trobar a partir de l'*offset* indicat a la secció NH corresponent.

Seccions NL

Llista dels identificadors d'arcs que van a parar als diferents nodes. S'accedeix a cadascun d'ells a partir de l'offset of i-th NL.

Per a cada **NODE** s'escriuen els identificadors dels arcs que conflueixen al node descrit. Cada ítem d'aquesta seqüència d'identificadors (tants com arcs conflueixin al node) ocupa 8 bytes (4 en la v. 1.x):



Arc id

Identificador de l'arc en el fitxer d'arcs que conflueix al node descrit. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

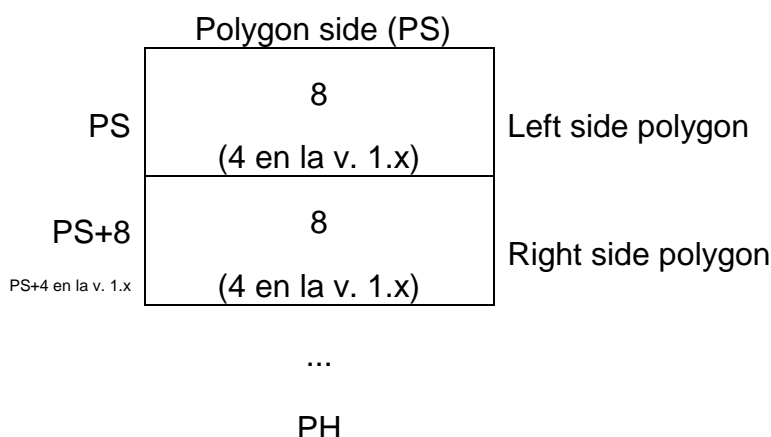
2.5. Fitxer de polígons .pol

El format de fitxers de polígons (POL) conté quatre seccions, descrites a continuació:

Secció TH. **Capçalera** comuna a tots els fitxers (**sizeof_TH**), anteriorment descrita. El camp "*File type*" correspon a la cadena a "POL".

Seccions PS. Contenen, per a cada **ARC** els identificadors dels polígons que són a l'esquerra i a la dreta (en aquest ordre) de l'arc descrit. La informació topològica del primer arc està escrita en l'offset **sizeof_TH**, que és on acaba sempre la capçalera comuna. Noteu, com s'ha explicat a la introducció, que una segona capa de polígons basada en la mateixa capa d'arcs farà referència a altres arcs, concretament a aquells que es necessitin per a ciclar els polígons de la segona capa. Aquesta capçalera té una grandària sizeof_PS, que val **16 bytes (8 bytes en la versió 1.x)**.

Descripció dels sizeof_PS bytes:



Left side polygon

Polígon situat al costat esquerre de l'arc descrit. S'usa un *unsigned __int64* (*unsigned __int32* en la versió 1.x).

Right side polygon

Polígon situat al costat dret de l'arc descrit. S'usa un *unsigned __int64* (*unsigned __int32* en la versió 1.x).

En la v. 2.0, els arcs que no participin en el ciclat tindran el valor a "0xFFFFFFFFFFFFFFFF" que és el màxim valor per a un *unsigned __int64*. En la v. 1.x, els arcs que no participin en el ciclat tindran el valor a "0xFFFFFFFF" que és el màxim valor per a un *unsigned __int32*.

Seccions PH. Per a cada **POLÍGON** s'escriu una capçalera de grandària sizeof_PH, que val **80 bytes (64 bytes en la versió 1.x)**. La primera capçalera es troba a sizeof_TH+sizeof_PS*n_arc i la resta a sizeof_TH+sizeof_PS*n_arc+sizeof_PH*id_pol.

Descripció dels sizeof_PH bytes de la capçalera d'un polígon:

Polygon Header (PH)		
PH	8	Bounding box: Minimum X
PH+8	8	Bounding box: Maximum X
PH+16	8	Bounding box: Minimum Y
PH+24	8	Bounding box: Maximum Y
PH+32	8 (4 en la v. 1.x)	Arcs count
PH+40	8 (4 en la v. 1.x)	Arcs in external rings count
PH+48	8 (4 en la v. 1.x)	Ring count
PH+56	8 (4 en la v. 1.x)	Offset of i-th PL
PH+64	8	Perimeter
PH+72	8	Area

Bounding box

Indica l'envolupant del polígon descrit en aquesta capçalera en l'ordre minX, maxX, minY, maxY. S'usa un *double* per a cada membre de l'envolupant.

Arcs count

Indica el nombre total d'arcs que formen el polígon descrit en aquesta capçalera. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Arcs in external rings count

Indica el nombre total d'arcs exteriors que formen el polígon descrit en aquesta capçalera. En la v. 2.0, si s'indica "0xFFFFFFFFFFFFFFFF" ("0xFFFFFFFF" en la v. 1.x) vol dir que no se sap res sobre quins arcs són interiors i quins són exteriors; en aquest cas el bit 0 de l'element corresponent de la VFG de Polygon Arc List (vegeu la secció PAL que ve a continuació) és sempre 0. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Ring count

Un **polipolígon** és una entitat poligonal que pot estar formada per més d'un anell. Aquest conjunt de bytes indica el nombre d'anells del polipolígon descrit en aquesta capçalera. En cas de polipolígons amb forats, els forats compten com a anells interiors del polipolígon. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Offset of i-th PL

Offset al primer arc del polígon descrit en aquesta capçalera. S'aconsella que sigui múltiple de 8. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

Perimeter

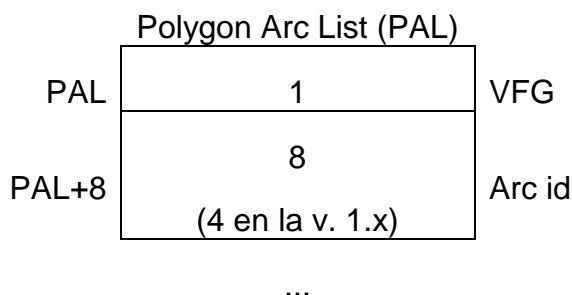
Perímetre del polígon descrit en aquesta capçalera, en el mateix sistema de referència que les coordenades. S'usa un *double*. Convé indicar que els perímetres sobre l'el·lipsoide també es calculen, però s'emmagatzemen a la taula principal (*P.dbf).

Area

Àrea del polígon descrit en aquesta capçalera, en el mateix sistema de referència que les coordenades. S'usa un *double*. Convé indicar que les àrees sobre l'el·lipsoide també es calculen, però s'emmagatzemen a la taula principal (*P.dbf).

Seccions PAL. Per a cada **POLIPOLÍGON** s'escriuen els arcs que el componen en una capçalera de grandària sizeof_PAL, que val **9 bytes (5 bytes en la versió 1.x)**.

Descripció dels sizeof_PAL bytes:



VFG

El **byte VFG** (Vora-Fi-Gir) és usat per a determinar característiques de l'arc que compon el polígon. Quan un bit està encès (valor 1) la propietat que defineix es considera que és VERITAT; altrament, FALS.

En la data d'aquest document estan definits els següents bits.

bit 0 (V): Indica si l'arc forma part d'un anell exterior (valor 1) o interior (0) del polipolígon.

bit 1 (F): Indica si amb l'arc actual acaba l'anell (valor 1) o si cal enllaçar amb un altre arc per tancar l'anell (valor 0). Per tant, si es compta el nombre d'uns (valor 1) d'un polipolígon aquest coincideix amb el nombre d'anells que formen el polipolígon.

bit 2 (G): Pot tenir dues interpretacions:

- a. Aquest booleà indica si el polígon en qüestió és al costat esquerre (valor 1) o al costat dret (valor 0) de l'arc segons el sentit de traçat dels vèrtexs en el fitxer ARC.
- b. En el cas de construir polígons explícits (no topològics) indica si s'ha de girar (valor 1) l'ordre de traçat dels vèrtexs escrit en el fitxer d'arcs per escriure aquest fragment d'anell en la successió de coordenades que descriuen tot l'anell explícitament, o si NO cal girar (valor 0) l'ordre del traçat dels vèrtexs escrit en el fitxer d'arcs per escriure aquest fragment d'anell; això és conseqüència que en els polígons explícits els anells s'han de construir fent que el polígon en si quedi a la dreta, cosa que permet que els anells exteriors quedin automàticament calculats amb àrea positiva i els interiors amb àrea negativa, i perquè es construeixin com s'acaba de descriure pot ser necessari haver d'invertir l'ordre de traçat dels vèrtexs.

NOTA: El polígon sempre queda a la dreta de les successió de coordenades que descriuen l'anell explícitament. Aquest criteri coincideix amb el criteri de l'ArcInfo. Això vol dir que els anells exteriors se ciclen en sentit horari i els anells interiors en sentit antihorari.

Arc id

Identificador de l'arc del fitxer d'arcs que és base d'aquest fitxer de polígons. S'usa un *unsigned __int64* (*unsigned __int32* en la v. 1.x).

2.6. Resum del format per a tots els tipus de fitxers

A continuació es mostra una taula resum de tots els formats dels fitxers vectorials estructurats del MiraMon.

Versió 1.x:

PNT		ARC		NOD		POL	
TH	48	TH	48	TH	48	TH	48
TL	16	AH	56	NH	8	PS	8
TL	...	AH	...	NH	...	PS	...
ZH	32	AL	16	NL	4	PH	64
ZD	24	AL	...	NL	...	PH	...
ZD	...	ZH	32			PAL	5
ZL	8	ZD	24			PAL	...
ZL	...	ZD	...				
		ZL	8				
		ZL	...				

Versió 2.0:

PNT		ARC		NOD		POL	
TH	56*	TH	56*	TH	56*	TH	56*
TL	16	AH	72*	NH	12*	PS	16*
TL	...	AH	...	NH	...	PS	...
ZH	32	AL	16	NL	8*	PH	80*
ZD	32*	AL	...	NL	...	PH	...
ZD	...	ZH	32			PAL	9*
ZL	8	ZD	32*			PAL	...
ZL	...	ZD	...				
		ZL	8				
		ZL	...				

* Marca que hi ha una diferència de la v. 1.x respecte de la versió 2.0. La diferència és deguda majoritàriament a dos fets: una ampliació en el nombre de

bytes per als identificadors d'objecte, que passen de 4 a 8 bytes, i una ampliació en els offsets on comencen les coordenades o els identificadors dels objectes descrits, que també passen de 4 a 8 bytes. A més, hi ha alguna reserva addicional d'espai que, alhora, permet una alineament a 8 bytes més generalitzat (no total perquè hi ha estructures, com la NH, que tenen una grandària inferior).

3. Algunes consideracions sobre els fitxers d'arcs i polígons

- Sempre hi ha un polígon que s'anomena polígon zero (o polígon universal). Aquest pren sentit en un fitxer amb la topologia garantida, però no en un fitxer de polígons explícits. El polígon zero està format per tots els arcs que formen anells de tots els altres polígons del fitxer sempre que aquests arcs no estiguin en contacte amb cap altre polígon. Per exemple, en un arxipèlag en què el mar fos el polígon zero, els diferents anells interiors serien les vores exteriors de les illes de l'arxipèlag. En el cas d'un fitxer format per un únic polígon amb un forat (no amb un altre polígon dintre del forat) el polígon zero té com a arcs tots els arcs del fitxer. Quan el polígon zero no té sentit topològic (típicament en capes de polígons explícits) aquest resta compost per zero arcs.
- En cas de ser un fitxer de polígons explícits, el polígon zero es documenta amb la capçalera completament plena amb 0 i no té secció PH ni PAL.
- Els fitxers de grups contenen polígons agrupats en grups de polígons. En aquests fitxers els identificadors dels objectes poden fer referència a polígons convencionals (amb o sense forats) o a grups de polígons. Cada objecte té una sola secció PH.
- L'ordre en que escrivim els arcs que componen un polígon (polipolígon) és:
 1. anell exterior
 2. anells interiors continguts en l'anell exterior anterior
 3. anell exterior (si n'hi ha algun altre)
 4. anells interiors continguts en l'anell exterior anterior.
 5. ...
- Cada vora interior compta en el recompte d'anells.
- El perímetre del polígon zero és la suma de longituds de totes les vores, amb signe positiu.
- L'àrea del polígon zero és la suma de les àrees de tots els polígons, assignant a la suma un signe negatiu.

Nota sobre polígons explícits: Un fitxer de **polígons explícits**, tant de grups com no, s'ha de definir com un fitxer POL amb les següents particularitats:

- Té desactivat el bit 0 i té activat el bit 5 del *flag* de la secció TH. El bit 3 d'aquest *flag* té el valor segons convingui.
- La secció PS (dels polígons de cada costat de cada arc) conté l'identificador gràfic del polígon a una banda i el polígon zero a

l'altra, segons el polígon explícit estigui al costat dret o a l'esquerre.

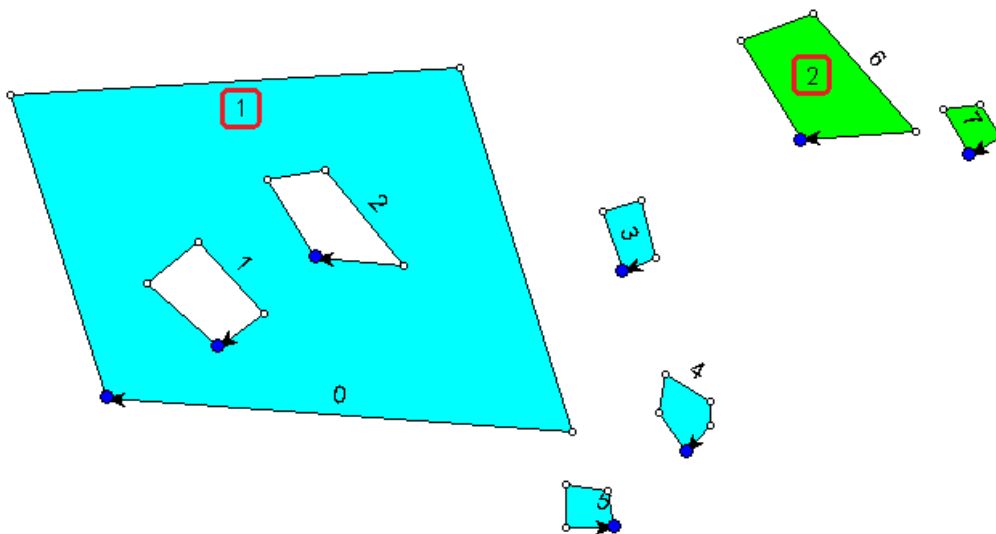
- El polígon zero està format per 0 arcs i no té secció PAL.

Un fitxer de **grups NO TOPOLÒGICS** (que admeten superposicions) s'ha definir com un fitxer POL amb les següents particularitats:

- Té desactivat el bit 0 i té activat el bit 3 del *flag* de la secció TH.
- La secció PS (dels polígons cada costat de cada arc) està plena amb 0xFFFFFFFFFFFFFFFF (màxim valor de tipus *unsigned __int64*) en la v. 2.0 (0xFFFFFFFF (màxim valor de tipus *unsigned __int32*) en la v. 1.x).
- El polígon zero està format per 0 arcs i no té secció PAL.

4. Exemple il·lustratiu de polígons complexos

El següent exemple il·lustra un dels casos més complexos: dos polígons, el primer amb dos forats i tres enclavaments i el segon amb cap forat i un enclavament.



El polígon a la part esquerra de l'exemple (blau) és el polígon d'identificador 1, perquè el polígon zero "no es veu" (no està pintat). El de la part dreta és el 2 (verd). L'identificador dels dos polígons està ressaltat amb una marca vermella. La resta dels identificadors corresponent a identificadors dels arcs, des del primer, amb valor 0, que encercla el polígon més gran, al darrer, amb valor 7 que està ubicat a la part dreta de la figura.

Per tal de clarificar aquest exemple, es descriuen alguns aspectes interessants a destacar:

- El nombre d'elements d'aquest polígon és 2, no 6 (que és el nombre d'anells). Això queda especificat en la capçalera del fitxer de polígons descrita en la secció TH, concretament en el byte 40 (i ocupa 8 bytes [4 en la v. 1.x]).
- El byte 7 (*flag*) de la capçalera del fitxer de polígons descrita en la secció PH tindrà activat, com a mínim, el bit 3.
- La secció PS, que determina quin polígon és al costat esquerre i dret té aquest aspecte: 01-10-10-01-01-01-02-02 (els guions només són ajudes visuals).
- En la secció PH (capçaleres dels polígons) cal destacar que la capçalera del polígon 1 tindrà 6 **Arcs count**, 4 **Arcs in external rings count** i 6 **Ring count** mentre que la capçalera del polígon 2 tindrà 2 **Arcs count**, 2 **Arcs in external rings count** i 2 **Ring count**.
- Finalment, la secció PAL tindrà el següent aspecte (es mostra **VFG** com un conjunt de tres bits):

0:	1-1-0, 0
	0-1-1, 1 (s'ha de girar perquè el sentit d'un forat és antihorari)
	0-1-1, 2 (s'ha de girar perquè el sentit d'un forat és antihorari)
	1-1-0, 3
	1-1-0, 4
	1-1-0, 5
1:	1-1-0, 6
	1-1-0, 7